LMU

LUDWIG-
MAXIMILIANS-
UNIVERSITÄT
MÜNCHEN

INSTITUT FÜR STATISTIK

Sebastian Kaiser, Dominik Träger, Friedrich Leisch

Generating Correlated Ordinal Random Values

Technical Report Number 94, 2011
Department of Statistics
University of Munich

http://www.stat.uni-muenchen.de

Generating Correlated Ordinal Random Values

Kaiser Sebastian   Traeger Dominik   Leisch Friedrich

Ludwig-Maximilians-Universität München, Department of Statstics, Ludwigstr. 33, 80539 München
firstname.lastname@stat.uni-muenchen.de

Abstract

Ordinal variables appear in many field of statistical research. Since working with simulated data is an accepted technique to improve models or test results there is a need for providing correlated ordinal random values with certain properties like marginal distribution and correlation structure. The present paper describes two methods for generating such values: binary conversion and a mean mapping approach. The algorithms of the two methods are described and some examples of the outcomes are shown.

Keywords: correlated ordinal values, marginal probabilities, correlation structure.

1 Introduction

A common method for testing a statistical model is the use of artificial data. A desired set of properties will be embedded in the dataset and then fitted models will be checked for the presence of these effects or how they behave under different experimental conditions. The generation of arbitrary multivariate normal random numbers is straightforward: draw values from the standard normal distribution, multiply with an appropriate root of the desired covariance matrix, and add the mean. Other distributions mostly call for more complicated solutions, because linear combinations in most cases do not preserve the type of distribution. Sampling count variables with a given correlation is described in Erhardt and Czado [2010]. For correlated binary data numerous methods have been proposed. For example Leisch et al. [1998] convert the desired covariance matrix for the binary data into a correlation matrix for normally distributed data. Therefrom normally distributed random numbers are drawn and binarised afterwards. For ordinal values only few suggestions can be found. Gange [1995] uses an iterative proportional fitting algorithm with pre specified probability structure of the marginals and pairwise and higher associations. Because of these higher order associations it becomes unpractical for large number of categories or variables. The method by Yu and Yuan [2004] works only for ordinal longitudinal data and needs an underlying regression model. Even more restrictions like independent and identical distribution among the variables are necessary for the method of Biswas [2004]. A more general solution can be found in Demirtas [2006]. His method relies on simulated binary variates as an intermediate step. Ordinal values are collapsed into binary ones, then corresponding binary correlations are computed in a way that ensures that reconversion delivers the original distribution properties. The first techniques (called binary conversion) proposed in the following is similar to the Demirtas [2006] approach, but has fewer restrictions on the kind of correlations used. Also an alternative approach will be presented which outperforms the binary conversion in many situations and is suitable in more situations.

In Section 2 we give an introduction to the generation of correlated multivariate binary variates following Leisch et al. [1998]. In Section 3 two techniques for generating multivariate ordinal variates are proposed. Section 4 shows some examples and compares the performanceces of the methods, and Section 5 ends the paper with some concluding remarks.

2 Generation of Correlated Binary Random Variates

In this section we deal with variables which take only binary values, typically encoded by $\{0,1\}$, and denoted by $A, B, \dots$ or $A_1, A_2, \dots$, respectively. Realizations of these random variables will be denoted by corresponding lower case letters. The distribution of a single variable A is fully determined by the value $p_A := \mathbb{P}(A=1)$, which is also the expectation of A, i.e., $\mathbb{E}A = p_A$. The variance is given by $\mathrm{Var}(A) = p_A(1-p_A)$.

Consider two binary random variables A and B which are not necessarily independent. Then the joint distribution of A and B is fully determined by p_A, p_B and either p_{AB}, $p_{A|B}$ or $p_{B|A}$ where

$$\begin{aligned} p_{AB} &:= \mathbb{P}(A=1, B=1) \\ p_{A|B} &:= \mathbb{P}(A=1|B=1) \\ p_{B|A} &:= \mathbb{P}(B=1|A=1) \end{aligned}$$

The remaining probabilities can easily be derived from Bayes Theorem.

This bivariate binary distribution can easily be generalized to the multivariate case, where $A = (A_1, \dots, A_d)' \in \{0,1\}^d$ is a vector with (possibly dependent) binary components. For a full description of an unrestricted distribution of A we need $2^d - 1$ parameters, e.g., the probabilities of all 2^d possible values of A (the last probability is determined by the condition that the sum equals 1).

A computationally fast method for generating samples from a binary vector $A = (A_1, \dots, A_d)$ is the following: Let $X = (X_1, \dots, X_d)$ be a d-dimensional normally distributed vector with mean μ and covariance matrix Σ. Normally distributed random variates can easily be transformed to binary values by componentwise thresholding: $a_i = 1 \iff x_i > 0$. Due to the construction

$$p_{A_i} = \mathbb{P}(A_i = 1) = \mathbb{P}(X_i > 0)$$

and

$$p_{A_i A_j} = \mathbb{P}(A_i = 1, A_j = 1) = \mathbb{P}(X_i > 0, X_j > 0),$$

where $\mathbb{P}(X_i > 0)$ depends, for fixed variances, only on μ_i whereas $\mathbb{P}(X_i > 0, X_j > 0)$ depends on μ_i, μ_j and on the correlation between X_i and X_j.

Let Y_i be a 1-dimensional normally distributed random variable with mean μ_i and unit variance. Hence,

$$\mathbb{P}(Y_i > 0) = \mathbb{P}((Y_i - \mu_i) > -\mu_i) = \mathbb{P}((Y_i - \mu_i) \leq \mu_i)$$

where the second equality holds, because $(Y_i - \mu_i)$ is normally distributed with zero mean. If we choose μ_i to be the p_{A_i}-quantile of the standard normal distribution and restrict all variances to 1, then $\mathbb{P}(Y_i > 0) = p_{A_i}$. The mean vector μ is determined by the desired marginal probabilities p_{A_i} for the components of A.

What is still missing is a relation between the covariance matrix Σ_b of the binary variables and the covariance matrix Σ of the normal distribution. By specifying a covariance matrix only pairwise relations between the components of the d-dimensional sample can be specified. In the following we will restrict ourself to the bivariate case for ease of notation.

The correlation coefficient r_{AB} of two binary random variables A and B can be written as

$$r_{AB} = \frac{p_{AB} - p_A p_B}{\sqrt{p_A(1-p_A)p_B(1-p_B)}} \tag{1}$$

such that

$$p_{AB} = r_{AB}\sqrt{p_A(1-p_A)p_B(1-p_B)} + p_A p_B. \tag{2}$$

If A and B are converted from two normal random variables X and Y as described above, then p_{AB} can be related to the normal distribution by

$$p_{AB} = \mathbb{P}(X > 0, Y > 0) = \mathbb{P}(\bar{X} > -\mu_X, \bar{Y} > -\mu_Y) = L(-\mu_X, -\mu_Y, \rho),$$

where $\bar{X} := X - \mu_X$ and $\bar{Y} := Y - \mu_Y$ have a standard bivariate normal distribution with correlation coefficient $\rho = \rho_{XY}$; and

$$L(h, k, \rho) := \mathbb{P}(\bar{X} \geq h, \bar{Y} \geq k) = \int_h^\infty \int_k^\infty \phi(x, y; \rho)dydx$$

with

$$\phi(x, y; \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \exp\left(-\frac{x^2 - 2\rho xy + y^2}{2(1-\rho^2)}\right)$$

being the density function of $(\bar{X}, \bar{Y})$.

The values of $L(h, k, \rho)$ are tabulated [see the references in Patel and Read, 1982, p. 293f] or can be obtained by numerical integration or Monte Carlo simulation [Leisch et al., 2009]. The complete algorithm is summarized in Table 2.

Note that not every positive definite matrix is a valid covariance matrix for binary data. So some conditions on the common probabilities and therefore on the correlation matrix should be checked before the algorithm draws random numbers. The conditions, besides $0 \leq p_{A_i} \leq 1$, are

$$\max(p_{A_i} + p_{A_j} - 1, 0) \leq p_{A_i A_j} \leq \min(p_{A_i}, p_{A_j}) \qquad i \neq j$$

and

$$p_{A_i} + p_{A_j} + p_{A_k} - p_{A_i A_j} - p_{A_i A_k} - p_{A_j A_k} \leq 1 \qquad , i \neq j,\ i \neq k,\ j \neq k.$$

These conditions are necessary but not sufficient for $d \leq 3$.

3 Generation of Correlated Ordinal Random Variates

Without loss of generality we want to generate ordinal variables A taking integer values $\{1, 2, \ldots, k\}$. The corresponding distribution is defined by probability vector

$$p_A = \begin{pmatrix} \mathbb{P}(A=1) \\ \mathbb{P}(A=2) \\ \vdots \\ \mathbb{P}(A=k) \end{pmatrix} = \begin{pmatrix} p_1 \\ p_2 \\ \vdots \\ p_k \end{pmatrix},$$

for notational reasons we also need the distribution function

$$f_A(a) = \begin{cases} p_1 & , \quad a = 1 \\ p_2 & , \quad a = 2 \\ \vdots & , \quad \vdots \\ p_k & , \quad a = k \end{cases} .$$

When generating random numbers for d ordinal variables $A_1, \ldots, A_d$ the user needs to specify the marginal probabilities p_{A_i}, $i = 1, \ldots, d$ and a positive semi-definite correlation matrix

$$\boldsymbol{C} = \begin{pmatrix} \mathrm{Cor}(A_1, A_1) & \mathrm{Cor}(A_1, A_2) & \ldots & \mathrm{Cor}(A_1, A_d) \\ \mathrm{Cor}(A_2, A_1) & \mathrm{Cor}(A_2, A_2) & \ldots & \mathrm{Cor}(A_2, A_d) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cor}(A_d, A_1) & \mathrm{Cor}(A_d, A_2) & \ldots & \mathrm{Cor}(A_d, A_d) \end{pmatrix}.$$

Higher order interactions will not be taken into account. Note that because we use $\{1, 2, \ldots, k\}$ as possible values, observed values correspond to ranks and Pearson and Spearman correlation are identical (only the latter makes sense for ordinal data in general).

In the case of binary random variates the region were two variables simultaneously equal one determines the correlation between them. There is a direct link between their common probabilities and their correlation. With more than two categories this region is not that clear cut. The correlation now rather depends on other regions i.e. the common probabilities $\mathbb{P}(A = a, B = b)$ $a = 1, ..., k_A$ $b = 1, ..., k_B$ as well. Considering this, two randomization methods that allow specification of means and correlations will be presented in this section.

3.1 The Binary Conversion Method

Demirtas [2006] used a simple splitting rule to convert the ordinal variables into binary variables. The lower half of the categories is represented by the binary 0 and the upper half by binary 1. A simulation study is carried out every time new random variables are drawn to identify the binary correlations. In the following we show a closed form solution for a very similar algorithm. The main idea is to draw binary random variables with the correct correlation structure, and conditional on the outcome of the binary variable convert an independent uniform random to an ordinal variable with the desired marginals and correlations.

Let $\tilde{A} := \dfrac{A-1}{k-1}$ denote a linear transformation of A to new outcome values $0, \frac{1}{k}, \ldots, \frac{k-1}{k}$. The expectation is given by

$$\mathbb{E}(\tilde{A}) = \sum_{a=1}^{k} \frac{a-1}{k-1} p_a \tag{3}$$

We also define a new binary variable A_b with distribution

$$f(A_b) := \begin{cases} 1 - \mathbb{E}(\tilde{A}) & , \quad A_b = 0 \\ \mathbb{E}(\tilde{A}) & , \quad A_b = 1 \end{cases}$$

such that $\mathbb{E}(\tilde{A}) = \mathbb{E}(A_b)$. In addition we get

$$\begin{aligned} \mathbb{E}(\tilde{A}\tilde{B}) &= \sum_{a=1}^{k_{\tilde{A}}} \sum_{b=1}^{k_{\tilde{B}}} \frac{a-1}{k_{\tilde{A}}-1} \frac{b-1}{k_{\tilde{B}}-1} \mathbb{P}(\tilde{A} = \frac{a-1}{k_{\tilde{A}}-1}, \tilde{B} = \frac{b-1}{k_{\tilde{B}}-1}) \\ &= \sum_{a=1}^{k_A} \sum_{b=1}^{k_B} \frac{a-1}{k_A-1} \frac{b-1}{k_B-1} \mathbb{P}(A = a, B = b) \\ &= \mathbb{P}(A_b = 1, B_b = 1) = \mathbb{E}(A_b B_b) \end{aligned}$$

and therefore

$$\begin{aligned} \mathrm{Cov}(\tilde{A}, \tilde{B}) &= \mathbb{E}(\tilde{A}\tilde{B}) - \mathbb{E}(\tilde{A})\mathbb{E}(\tilde{B}) = \mathbb{E}(A_b B_b) - \mathbb{E}(A_b)\mathbb{E}(B_b) \\ &= \mathrm{Cov}(A_b, B_b). \end{aligned} \tag{4}$$

Using $\mathrm{Var}(A_b) = \mathbb{E}(\tilde{A})(1 - \mathbb{E}(\tilde{A}))$ we get

$$\mathrm{Var}(\tilde{A}) = \sum_{a=1}^{k} (\frac{a-1}{k-1} - \mathbb{E}(\tilde{A}))^2 p_a$$

and analogously for $\mathrm{Var}(\tilde{B})$. Due to the linearity of the conversion $\mathrm{Cor}(\tilde{A}, \tilde{B}) = \mathrm{Cor}(A, B)$. The function that maps the desired correlation $\mathrm{Cor}(A, B)$ on the binarised correlation $\mathrm{Cor}(A_b, B_b)$ is a straight line passing through the origin and with slope m that depends only on the probability vectors p_A and p_B:

$$\mathrm{Cor}(\tilde{A}, \tilde{B}) = \mathrm{Cor}(A, B) = m\mathrm{Cor}(A_b, B_b) \tag{5}$$

For four examples of probability vectors for two variables this is shown in Figure 1.

Combining (5) and (4) gives

$$\begin{aligned} m^{-1} &= \frac{\mathrm{Cor}(A_b, B_b)}{\mathrm{Cor}(\tilde{A}, \tilde{B})} = \frac{\dfrac{\mathrm{Cov}(A_b, B_b)}{\sqrt{\mathrm{Var}(A_b)\mathrm{Var}(B_b)}}}{\dfrac{\mathrm{Cov}(\tilde{A}, \tilde{B})}{\sqrt{\mathrm{Var}(\tilde{A})\mathrm{Var}(\tilde{B})}}} = \sqrt{\frac{\mathrm{Var}(\tilde{A})\mathrm{Var}(\tilde{B})}{\mathrm{Var}(A_b)\mathrm{Var}(B_b)}} \\ &= \sqrt{m_A m_B}, \end{aligned}$$

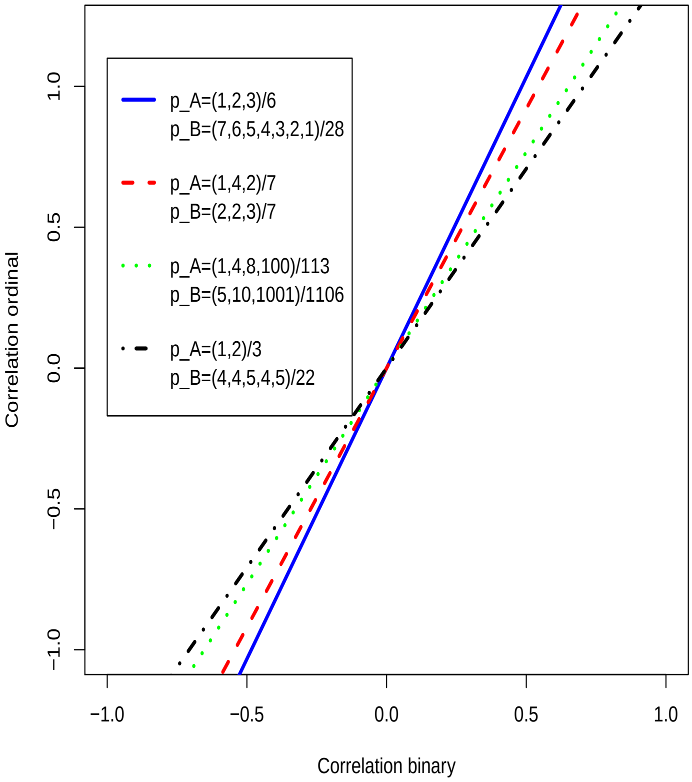

Figure 1: Linear transformation functions. The m-factors to translate ordinal correlation specifications to binary correlations.

with $m_A = \mathrm{Var}(\tilde{A})/\mathrm{Var}(A_b)$ and $m_B = \mathrm{Var}(\tilde{B})/\mathrm{Var}(B_b)$.
Using

$$\begin{aligned}\sum_{a=1}^{k_A}(\frac{a-1}{k_A-1} - \mathbb{E}(\tilde{A}))^2\mathbb{E}(\tilde{A}) &= \mathbb{E}(\tilde{A})(1-\mathbb{E}(\tilde{A})) + \sum_{a=1}^{k_A}(\frac{a-1}{k_A-1})^2\mathbb{E}(\tilde{A}) - \mathbb{E}(\tilde{A}) \\ &= \sum_{a=1}^{k_A}(\frac{a-1}{k_A-1})^2\mathbb{E}(\tilde{A}) - 2\mathbb{E}(\tilde{A})^2 + \mathbb{E}(\tilde{A})^2 \\ &= -\mathbb{E}(\tilde{A})^2 + \sum_{a=1}^{k_A}(\frac{a-1}{k_A-1})^2\mathbb{E}(\tilde{A})\end{aligned}$$

we get

$$\mathrm{Var}(\tilde{A}) = \mathrm{Var}(A_b) + \mathbb{E}(\tilde{A}^2) - \mathbb{E}(A_b).$$

The conditional distribution of A given A_b is

$$f(A|A_b) = \begin{cases} f(A|A_b = 0) =: f_0(A) \\ f(A|A_b = 1) =: f_1(A) \end{cases}$$

For $A_b = 1$ the conditional distribution $f_1(A)$ is simply

$$f_1(A) = \frac{\frac{a-1}{k-1}p_a}{\mathbb{E}(\tilde{A})} = \frac{(a-1)p_a}{\sum_{l=2}^{k}(l-1)p_l},$$

for $A_b = 0$ we can use

$$\begin{aligned} \mathbb{P}(A_b = 0) &= 1 - \mathbb{E}(\tilde{A}) \\ &= 1 - \sum_{a=1}^{k} \frac{a-1}{k-1}p_a = 1 - \frac{1}{k-1}(\mathbb{E}(A) - 1) \\ &= \frac{k - \mathbb{E}(A)}{k-1} = \frac{\sum_{a=1}^{k} kp_a - \sum_{a=1}^{k} ap_a}{k-1} = \\ &= \sum_{a=1}^{k} \frac{k-a}{k-1}p_a = \sum_{a=1}^{k-1} \frac{k-a}{k-1}p_a, \end{aligned}$$

to obtain

$$f_0(A) = \frac{\frac{k-a}{k-1}p_a}{1 - \mathbb{E}(\tilde{A})} = \frac{(k-a)p_a}{\sum_{l=1}^{k-1}(k-l)p_l}.$$

The resulting cumulative distribution functions are therefore

$$F_0(A) = \frac{\sum_{l=1}^{a} \frac{k-l}{k-1}p_l}{1 - \mathbb{E}(\tilde{A})}$$

and

$$F_1(A) = \frac{\sum_{l=2}^{a} \frac{l-1}{k-1}p_l}{\mathbb{E}(\tilde{A})}.$$

The final algorithm is to draw binary variables A_b with a certain correlation structure. In addition we independently draw from the uniform distribution $U(0, 1)$ and use the inversion method with $F_1(A)$ and $F_0(A)$ to obtain ordinal values. The binary variables A_b shift the distribution of A to the left or right to get correlations, the particular choice of A_b guarantees that the overall marginal probabilities are still correct. The whole algorithm is summarized in Table 3

Figure 1 shows that not all correlations can be calculated because the binary correlations are restricted to $[-1, 1]$. Hence, the correlation range of the algorithm is smaller than that of the method in Demirtas [2006]. But while they use simulation runs we have an analytical solution for the transformation which leads to far shorter run times. Since range may be more important than speed, the next section gives an alternative approach with broader range.

3.2 The Mean Mapping Method

Our mean mapping method to generate ordinal random numbers with a given correlation structure generalizes the concepts of Leisch et al. [1998] from the binary to the ordinal case. Let X again be a random variable with standard normal distribution $N(0,1)$. To get an ordinal variable with cumulative distribution F we cut X at the $F(a)$-quantiles q of the standard normal distribution:

$$\mathbb{P}(A=a)=\mathbb{P}(q_{F_A(a-1)}<X<q_{F_A(a)}) \qquad a=1,\dots,k_A \qquad X\sim N(0,1), \quad (6)$$

Figure 2 shows an example for $k=4$ categories.

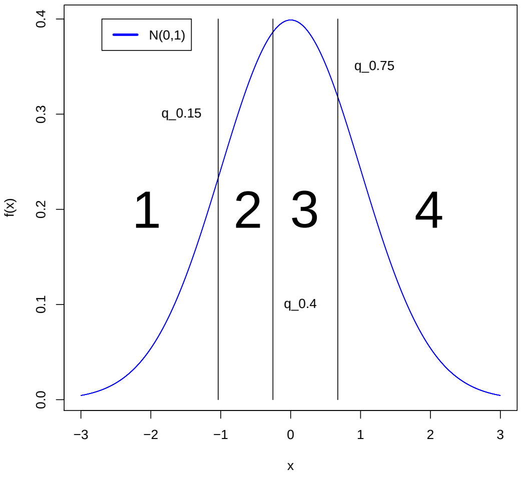

Figure 2: Thresholding the normal distribution.
$\boldsymbol{p}_A=(0.15 \quad 0.25 \quad 0.35 \quad 0.25)^\mathsf{T} \Rightarrow \boldsymbol{q}_A\approx(-1.04 \quad -0.25 \quad 0.67 \quad +\infty)^\mathsf{T}$

Let A and B be two ordinal variables obtained by cutting X and Y, respectively. The joint probabilities can then be written as

$$\begin{aligned}&\mathbb{P}(A=a,B=b) && (7)\\ &= F_{AB}(a,b)-F_{AB}(a-1,b)-F_{AB}(a,b-1)+F_{AB}(a-1,b-1)\\ &= \Phi_{XY}(q_{F_A(a)},q_{F_B(b)},\rho_{XY})-\Phi_{XY}(q_{F_A(a-1)},q_{F_B(b)},\rho_{XY}) && (8)\\ &\quad -\Phi_{XY}(q_{F_A(a)},q_{F_B(b-1)},\rho_{XY})+\Phi_{XY}(q_{F_A(a-1)},q_{F_B(b-1)},\rho_{XY})\end{aligned}$$

with q being a quantile of the univariate standard normal distribution and $\mathbb{P}(X<h,Y<k)=\Phi_{XY}(h,k,\rho_{XY})$ the bivariate standard normal distribution function with correlation coefficient ρ_{XY}. Equation (9) links probabilities $\mathbb{P}(A=a,B=b)$ to ρ_{XY}. For the binary case $\mathbb{P}(A=1,B=1)=\mathbb{E}(AB)$ defines the whole distribution. Hence, the natural generalization for the ordinal case would be to evaluate the relationsship between $\mathbb{E}(AB)$ and ρ_{XY} on a regular grid and interpolate the results. For this we would need to specify the complete joint distribution of F_{AB}. By rearranging terms we can find a scalar (called τ below) which only depends on the marginal distribution of A and B and the desired correlation $\text{Cor}(A,B)$.

The expectaion of AB is defined as

$$\begin{aligned}\mathbb{E}(AB) &= \sum_{a=1}^{k_A}\sum_{b=1}^{k_B} ab\,\mathbb{P}(A=a,B=b) \\ &= \sum_{a=1}^{k_A}\sum_{b=1}^{k_B} ab\,\big(F_{AB}(a,b)-F_{AB}(a-1,b) \\ &\quad -F_{AB}(a,b-1)+F_{AB}(a-1,b-1)\big) \\ &= \sum_{a=1}^{k_A}\sum_{b=1}^{k_B} m_{ab}F_{AB}(a,b). \qquad (9)\end{aligned}$$

By simple algebra we get the multiplicities m_{ab} as

$$\begin{array}{lrll} m_{ab}=ab-a(b+1)-(a+1)b+(a+1)(b+1)= & 1, & a<k_A & b<k_B \\ m_{ab}=a[b-(b+1)]=-a= & -k_A, & a=k_A & b<k_B \\ m_{ab}=b[a-(a+1)]=-b= & -k_B, & a<k_A & b=k_B \\ m_{ab}=ab= & k_Ak_B, & a=k_A & b=k_B \end{array} \qquad (10)$$

Combining Equations (9) and (10) gives

$$\begin{aligned}\mathbb{E}(AB) &= \sum_{a=1}^{k_A-1}\sum_{b=1}^{k_B-1} F_{AB}(a,b) - k_B\sum_{a=1}^{k_A-1}F_A(a) \\ &\quad -k_A\sum_{b=1}^{k_B-1}F_B(b)+k_Ak_B.\end{aligned}$$

We use the first term of this equation as proxy τ which will be linked to ρ_{XY}. Rearranging terms in the usual definition of the correlation gives

$$\begin{aligned}\tau_{AB} = &\sum_{a=1}^{k_A-1}\sum_{b=1}^{k_B-1} F_{AB}(a,b) \\ = &\quad \mathrm{Cor}(A,B)\sqrt{\mathrm{Var}(A)\mathrm{Var}(B)}+\mathbb{E}(A)\mathbb{E}(B) \\ &\quad -k_Ak_B+k_A\sum_{b=1}^{k_B-1}F_B(b)+k_B\sum_{a=1}^{k_A-1}F_A(a),\end{aligned}$$

which depends only on the marginal distribution of A and B and correlation $\mathrm{Cor}(A,B)$. We now evaluate the relationship between ρ_{XY} and τ_{AB} on a regular grid and interpolate results. Inverting this relationsship gives the necassary ρ_{XY} for given τ_{AB}. Drawing random numbers now amounts to drawing bivariate normal variates with zero mean, unit variance and correlation ρ_{XY}. These are then cut at quantiles defined by the marginal distributions of A and B, respectively. Generalization to more than two ordinal variates is again straightforward. The complete algorithm for the mean mapping method can be found in Table 4.

4 Simulation and Comparison

For comparison of the two methods we generated random ordinal values from both methods and compared the results with respect to runtime and precision.

As the restrictions on the correlation matrix are stronger for the binary conversion method than for the mean mapping method, matrices are chosen which are feasible for both methods. As dimensions d and number of categories k we used 3, 6 and 9 in both cases. One million random values were drawn for each algorithm with each of the 9 setups.

4.1 Performance

The runtime of the algorithms is depicted in Figure 3. It can be seen that the runtime of the binary conversion method is very low even for the case with 9 variables and 9 categories. The runtime of the mean mapping method depends on both, the numbers of categories and the number of variables.

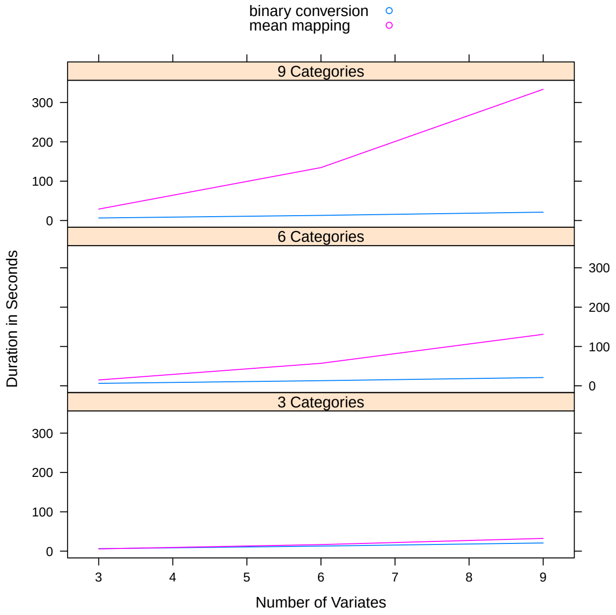

Figure 3: Runtime of binary and mean mapping method.

4.2 Accuracy

Figures 4, 5 and 6 give information about how exact the methods generate random numbers. For this purpose the following quantities were calculated: *Average absolute distance of correlation matrix entries:*

$$\mu_C = \frac{1}{q^2} \sum_{i=1}^{q} \sum_{j=1}^{q} |\boldsymbol{C}_{[i,j]} - \hat{\boldsymbol{C}}_{[i,j]}|$$

Maximum absolute distance of correlation matrix entries:

$$m_C = \max_{i,j}(|\boldsymbol{C}_{[i,j]} - \hat{\boldsymbol{C}}_{[i,j]}|)$$

Average absolute distance of probability vector entries:

$$\mu_P = \sum_{i=1}^{q} \sum_{a_i=1}^{k_{A_i}} |\boldsymbol{P}_{[i,a_i]} - \hat{\boldsymbol{P}}_{[i,a_i]}|$$

with $\hat{\boldsymbol{C}}$ the empirical correlation matrix computed from the observed random numbers and $\hat{\boldsymbol{P}}$ relative frequencies of the cases computed from the observed random numbers.

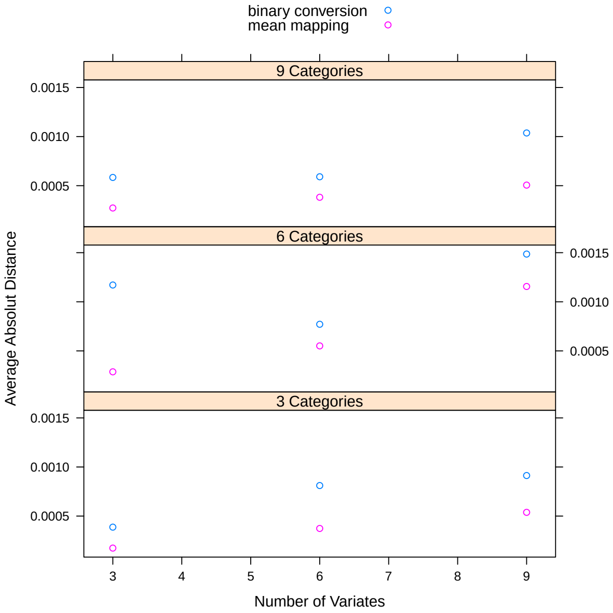

Figure 4: Average absolute differences of sample and input correlations.

Figure 4 shows that all values for μ_C do not exceed 0.003 with the largest average distance being at $\mu_C = 0.002967$ which is a good result. One can also note that the mean mapping method is the numerically most stable. A similar result is indicated by figure 5 which presents the m_C values. Again both methods are similar, but the mean mapping method is better.

Figure 6 shows that both methods have similar low values for μ_P, which had to be expected because all methods use a categorization of the normal distribution which is analytically exact. One can also see that for more categories μ_P does slightly shrink, which is what we can expect due to the increased number of observations $\hat{\boldsymbol{P}}_{[i,a_i]}$ that enter the formula. Summarizing the results, μ_C, m_C and μ_P show that both methods have sufficient precision for most practical applications.

4.3 Comparison with Demirtas

In Demirtas [2006] different setups were use to show the flexibility of the algorithm. In this section we show that the mean mapping approach can cover all

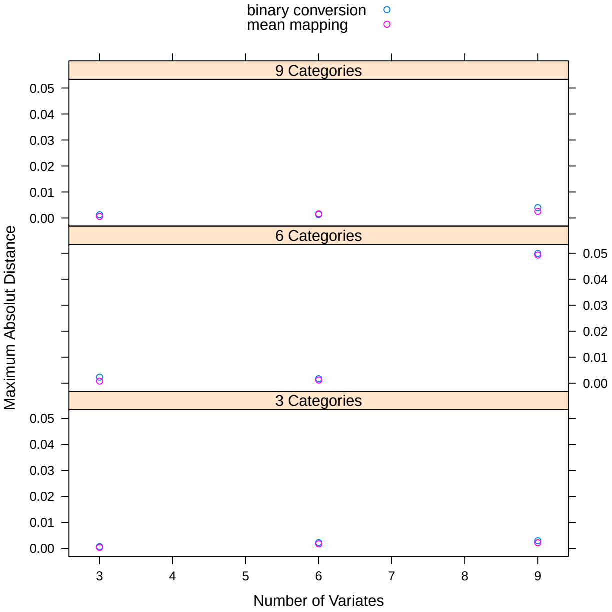

Figure 5: Maximum absolute differences of sample and input correlations.

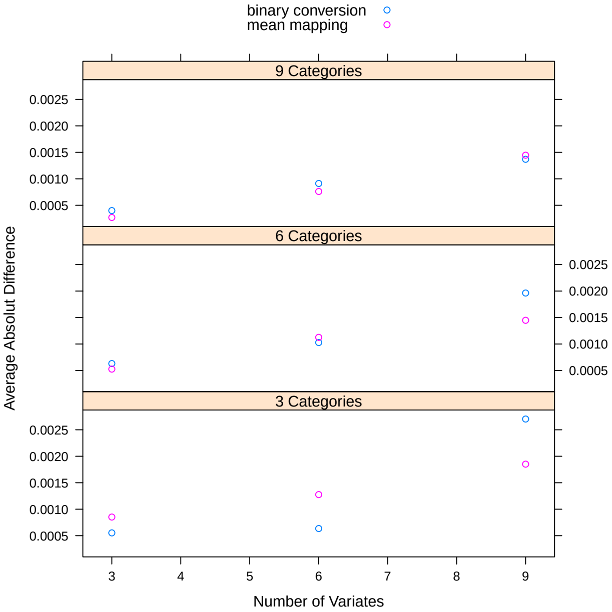

Figure 6: Average absolute differences of sample and input probabilities.

	Cor	Mat	1	Cor	Mat	1	Cor	Mat	3	
	A1	A2	A3	A1	A2	A3	A1	A2	A3	
A1	1	0.4	0.3	1	0.5	0.25	1	0.7	0.7	A1
A2	0.4	1	0.4	0.5	1	0.5	0.7	1	0.7	A2
A3	0.3	0.4	1	0.25	0.5	1	0.7	0.7	1	A3

Table 1: Three example correlation matrices

these setups and can also extend these setups to higher correlations. Table 1 contains two examples of correlation matrices which were used by Demirtas [2006] and a third matrix which is not feasible for his method. As marginal probabilities we used

$$P_{A1} = \begin{pmatrix} 0.05 \\ 0.25 \\ 0.55 \\ 0.15 \end{pmatrix}, p_{A2} = \begin{pmatrix} 0.10 \\ 0.10 \\ 0.10 \\ 0.70 \end{pmatrix}, p_{A3} = \begin{pmatrix} 0.20 \\ 0.15 \\ 0.25 \\ 0.40 \end{pmatrix}$$

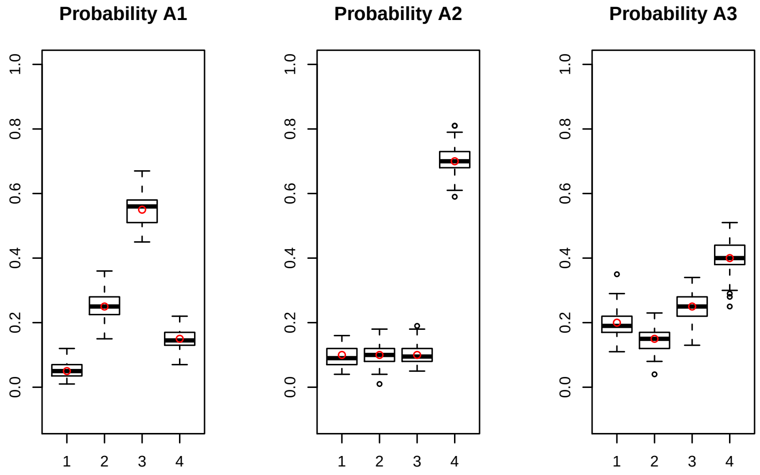

Figure 7: Boxplot of frequency of 100 simulation runs with Correlation matrix 1. Red circles show input probabilities.

Figure 7 shows the frequencies of 100 simulation runs were 100 random ordinal variates were drawn. The red circles represent the desired values, which are close to the median of the observed values in each case. Figure 8 shows the three values of the upper triangle of the observed correlation matrices with the red circles again representing the desired correlation. It can be seen, that the algorithm works quite good in all three scenarios.

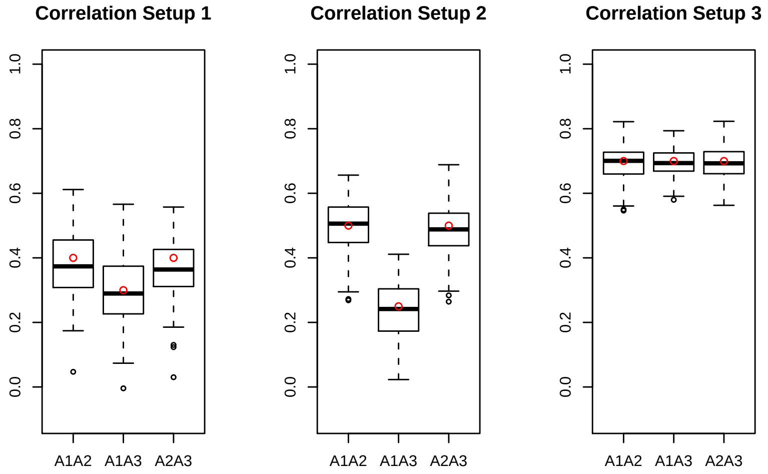

Figure 8: Boxplot of correlations of 100 simulation runs. Red circles show input correlations.

5 Conclusions

In the paper we presented two new methods for generating ordinal values with given correlation structure. The binary method is very fast but has the disadvantage that the set of feasible correlation matrices is limited by the algorithm. The mean mapping method overcomes this problem and is as accurate as the binary solution at the price of longer runtime. With more and more statistical models working on samples of ordinal values were normality cannot be assumed, the presented method are valuable tools for simulation studies. A freely available open source implementation for the statistical computing environment R [R Development Core Team, 2010] is described in the appendix.

References

Atanu Biswas. Generating correlated ordinal categorical random samples. *Statistics & Probability Letters*, 70(1):25–35, October 2004. URL `http://ideas.repec.org/a/eee/stapro/v70y2004i1p25-35.html`.

Hakan Demirtas. A method for multivariate ordinal data generation given marginal distributions and correlations. *Journal of Statistical Computation and Simulation*, 76(11):1017–1025, 2006. URL `http://www.informaworld.com/10.1080/10629360600569246`.

Vinzenz Erhardt and Claudia Czado. *Sampling Count Variables with specified Pearson Correlation - a Comparison between a naive and a C-vine Sampling Approach*. World Scientific Publishing Company, 2010.

Stephen J. Gange. Generating multivariate categorical variates using the iterative proportional fitting algorithm. *The American Statistician*, 49(2):134–138, 1995. ISSN 00031305. URL `http://www.jstor.org/stable/2684626`.

Sebastian Kaiser and Friedrich Leisch. *orddata: Generation of Artificial Ordinal and Binary Data*, 2010. R package version 0.1.

Friedrich Leisch, Andreas Weingessel, and Kurt Hornik. On the generation of correlated artificial binary data. Technical Report 13, SFB Adaptive Information Systems and Modelling in Economics and Management Science, Wirtschaftsuniversität Wien, Augasse 2-6, A-1090 Wien, Austria, 1998.

Friedrich Leisch, Andreas Weingessel, and Kurt Hornik. *bindata: Generation of Artificial Binary Data*, 2009. URL `http://CRAN.R-project.org/package=bindata`. R package version 0.9-17.

Jagdish K. Patel and Campbell B. Read. *Handbook of the Normal Distribution*, volume 40 of *Statistics: Textbooks and Monographs*. Marcel Dekker, Inc., New York and Basel, 1982.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

Kai Fun Yu and Weishi Yuan. Regression models for unbalanced longitudinal ordinal data: computer software and a simulation study. *Computer Methods and Programs in Biomedicine*, 75(3):195 – 200, 2004. ISSN 0169-2607. doi: DOI: 10.1016/j.cmpb.2004.02.006. URL `http://www.sciencedirect.com/science/article/B6T5J-4CB0P6X-1/2/1bb9a09a8193f301a57eb309e97b045c`.

Step	Expression
1	Calculate the probabilities $h^{t_1}_{t_2,t_3} := \int_h^\infty \int_k^\infty \phi_{X_1X_2}(g_{t_2}, g_{t_3}, \boldsymbol{\Sigma}_{t_1})dxdy$ with $(X_1, X_2) \sim N(\mathbf{0}, \boldsymbol{\Sigma}_{t_1})$ and covariance matrix $\boldsymbol{\Sigma}_{t_1} = \begin{pmatrix} 1 & g_{t_1} \\ g_{t_1} & 1 \end{pmatrix}$ where $g_{t_1} = \dfrac{t_1}{20}$ and $t_1 = -20, -19, \ldots, 20$, $g_{t_2} = \dfrac{t_2}{20}$ and $t_2 = 0, 1, \ldots, 20$ and $g_{t_3} = \dfrac{t_3}{20}$ and $t_3 = 0, 1, \ldots, 20$ receiving grid $\boldsymbol{g}_{1,2,3}$.
2	Fit a function $f_{h\vert g_{t_2},g_{t_3}}(h) : h \rightarrow g_{t_1}$ to grid $\boldsymbol{g}_{1,2,3}$.
3	Set $h^* = \text{Cor}(A_1, A_2)\sqrt{\text{Var}(A_1)\text{Var}(A_2)} + \mathbb{E}(A_1)\mathbb{E}(A_2)$ and calculate the correlation coefficient $f_{h\vert g_{t_2},g_{t_3}}(h^*) = \text{Cor}(X_1, X_2).$
4	Repeat step 3 for all combinations (i, j) of variables, receiving $\boldsymbol{C}^{bin} = \begin{pmatrix} \text{Cor}(X_1, X_1) & \text{Cor}(X_1, X_2) & \ldots & \text{Cor}(X_1, X_d) \\ \text{Cor}(X_2, X_1) & \text{Cor}(X_2, X_2) & \ldots & \text{Cor}(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cor}(X_d, X_1) & \text{Cor}(X_d, X_2) & \ldots & \text{Cor}(X_d, X_d) \end{pmatrix},$ which is the multivariate normal correlation matrix.
5	Sample n times from the d-dimensional normal distribution with covariance matrix $\boldsymbol{C}^{bin}$ and mean vector $\boldsymbol{\mu}^{bin} = \begin{pmatrix} -q_{p(A_1)} \\ -q_{p(A_2)} \\ \vdots \\ -q_{p(A_d)} \end{pmatrix}$ receiving the $n \times d$ sample matrix $\boldsymbol{S}^{bin}$.
6	Reconvert $\boldsymbol{S}^{bin}$ to the ordinal sample matrix $\boldsymbol{S}$ using $\boldsymbol{S}_{[ei]} = \begin{cases} 1 & , 0 \leq \boldsymbol{S}^{bin}_{[ei]} \\ 0 & , 0 > \boldsymbol{S}^{bin}_{[ei]} \end{cases}$

Table 2: Generation of multivariate binary random numbers via Leisch et al. [1998]

Step	Statement
1	Calculate the weighted mean $\mu_1 := \mathbb{E}(A_1^b) = \sum_{a=1}^{k_1} \frac{a-1}{k_1-1} p_a$.
2	Calculate the binarised variance $\mathrm{Var}(A_1^b) = \mathbb{E}(\tilde{A})(1 - \mathbb{E}(\tilde{A}))$.
3	Calculate the ordinal variance $\mathrm{Var}(\tilde{A}_1) = \sum_{a=1}^{k_1} (\frac{a-1}{k_1-1} - \mathbb{E}(\tilde{A}))^2 p_a$.
4	Calculate the slope $m_1 := \mathrm{Var}(\tilde{A}_1)/\mathrm{Var}(A_1^b)$.
5	Do steps 1-4 for each of the d variables receiving $\boldsymbol{\mu}^b = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \vdots \\ \mu_d \end{pmatrix} \quad and \quad \boldsymbol{m} = \begin{pmatrix} m_1 \\ m_2 \\ m_3 \\ \vdots \\ m_d \end{pmatrix}.$
6	Calculate the new binary correlation matrix via $\mathrm{Cor}(A_i^b, A_j^b) = \begin{cases} \mathrm{Cor}(A_i, A_j)/\sqrt{m_i m_j} & , \quad i \neq j \\ 1 & , \quad i = j \end{cases}$ getting $\boldsymbol{C}^b = \begin{pmatrix} \mathrm{Cor}(A_1^b, A_1^b) & \mathrm{Cor}(A_1^b, A_2^b) & \dots & \mathrm{Cor}(A_1^b, A_d^b) \\ \mathrm{Cor}(A_2^b, A_1^b) & \mathrm{Cor}(A_2^b, A_2^b) & \dots & \mathrm{Cor}(A_2^b, A_d^b) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cor}(A_d^b, A_1^b) & \mathrm{Cor}(A_d^b, A_2^b) & \dots & \mathrm{Cor}(A_d^b, A_d^b) \end{pmatrix}$
7	Sample n times from d-dimensional binary distribution with correlation matrix of the binary distribution $\boldsymbol{C}^b$ and mean vector $\boldsymbol{\mu}^b$ getting the $n \times d$ binary sample matrix $\boldsymbol{S}^b$
8	Draw from $U(0,1)$ md times receiving $m \times d$ matrix $\boldsymbol{U}$.
9	Reconvert $\boldsymbol{S}^b$ for each variable with originally more than two categories to the ordinal sample matrix $\boldsymbol{S}$ using the samples from 7 and 8 and the assignment
.	$\boldsymbol{S}_{[ei]} = k : \begin{cases} F_0(k-1) < \boldsymbol{U}_{[ei]} < F_0(k) & , \quad if\ \boldsymbol{S}_{[ei]}^b = 0 \\ F_1(k-1) < \boldsymbol{U}_{[ei]} < F_1(k) & , \quad if\ \boldsymbol{S}_{[ei]}^b = 1 \end{cases}$ for $k \in \{1, 2, \dots, k_i\}$ with cumulative distribution functions $F_0(A) = \frac{\sum_{l=1}^{a} \frac{k-l}{k-1} p_l}{1 - \mathbb{E}(\tilde{A})}$ and $F_1(A) = \frac{\sum_{l=2}^{a} \frac{l-1}{k-1} p_l}{\mathbb{E}(\tilde{A})}$ for each entry $\boldsymbol{S}_{[ei]}$ independently.

Table 3: Generating multivariate ordinal random numbers via binary conversion method

Step	Expression
1	Calculate the probability $h^t_{1,2} := \sum_{a_1=1}^{k_1-1} \sum_{a_2=1}^{k_2-1} \mathbf{\Phi}_{X_1 X_2}(q_{F_{A_1}(a_1)}, q_{F_{A_2}(a_2)})$ with $(X_1, X_2) \sim N(\mathbf{0}, \mathbf{\Sigma}_t^{1,2})$ and covariance matrix $\mathbf{\Sigma}_t^{1,2} = \begin{pmatrix} 1 & g_t \\ g_t & 1 \end{pmatrix}$ where $g_t = \dfrac{t}{100}$ and $t = -100, \dots, 100$, receiving grid $\boldsymbol{g}_{1,2}$.
2	Fit a function $f_{1,2}(h) : h \to g$ to grid $\boldsymbol{g}_{1,2}$.
3	Set $h^* = \mathrm{Cor}(A_1, A_2)\sqrt{\mathrm{Var}(A_1)\mathrm{Var}(A_2)} + \mathbb{E}(A_1)\mathbb{E}(A_2)$ $-k_1 k_2 + k_1 \sum_{a_2=1}^{k_2-1} F_{A_2}(a_2) + k_2 \sum_{a_1=1}^{k_1-1} F_{A_1}(a_1)$ and calculate the correlation coefficient $f_{1,2}(h^*) = \mathrm{Cor}(X_1, X_2).$
4	Repeat steps 1-3 for all combinations (i, j) of variables, receiving $\boldsymbol{C}^a = \begin{pmatrix} \mathrm{Cor}(X_1, X_1) & \mathrm{Cor}(X_1, X_2) & \dots & \mathrm{Cor}(X_1, X_d) \\ \mathrm{Cor}(X_2, X_1) & \mathrm{Cor}(X_2, X_2) & \dots & \mathrm{Cor}(X_2, X_d) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cor}(X_d, X_1) & \mathrm{Cor}(X_d, X_2) & \dots & \mathrm{Cor}(X_d, X_d) \end{pmatrix},$ which is the multivariate normal correlation matrix.
5	Sample n times from the d-dimensional normal distribution with covariance matrix $\boldsymbol{C}^a$ and mean vector $\boldsymbol{\mu}^a = \mathbf{0}$ receiving the $n \times d$ sample matrix $\boldsymbol{S}^a$.
6	Reconvert $\boldsymbol{S}^a$ to the ordinal sample matrix $\boldsymbol{S}$ using $\boldsymbol{S}_{[ei]} = k : \{F_{A_i}(k-1) < \Phi(\boldsymbol{S}^a_{[ei]}) < F_{A_i}(k)\}$ for $k \in \{1, 2, \dots, k_i\}$ and Φ being the univariate standard normal distribution function.

Table 4: Generation of multivariate ordinal random numbers via the mean mapping method.

Appendix

We provide an implementation of all the methods used in this paper as add on package orddata [Kaiser and Leisch, 2010] for R [R Development Core Team, 2010]. It extends package bindata [Leisch et al., 2009] which contains the method of Leisch et al. [1998] for drawing correlated binary data, and will eventually replace it. In this appendix we give a small manual how to use the methods for the simulation study used in this paper.

The package can be downloaded from R-Forge and loaded into R using

```
> install.packages("orddata", repos = "http://R-Forge.R-project.org")
> library("ordata")
```

The main function of the package is `rmvord()`, which returns `n` observations with given marginal probabilities `probs` and correlation structure `Cor` using the mean mapping algorithm. `probs` is a list of probabilities for the variables where length of list equals number of variables and the length of the probabilities equals the number of items. The `probs` list for the example in section 4.3 looks like this

```
> probs1 <- list(c(5, 25, 55, 15)/100, c(10, 10, 10, 70)/100,
+     c(20, 15, 25, 40)/100)
```

The first correlation matrix of Table 1 can be specified by

```
> Cor1 <- matrix(c(1, 0.4, 0.3, 0.4, 1, 0.4, 0.3, 0.4,
+     1), 3, 3)
```

To draw `n =100` observation one then has to call

```
> rmvord(n = 100, probs = probs1, Cor = Cor1)
```

If a faster production of correlated ordinal values is needed and the restrictions to the correlation matrix do not apply the function

```
> rmvord_b(, n = 100, probs = probs1, Cor = Cor1)
```

does the same using the faster binary conversion method described in section 3.

For further details and examples, please see the package manual.